Design of Multi Agent Autonomous Workflow Systems Using Agentic AI Frameworks

[1]Harish Apuri
Software engineer and IT Induct INC.
Charlotte, North Carolina. USA.
hapuri1029@gmail.com

[2]Charani yepuri
student
Hyderabad, India.
Charaniy04@gmail.com.

Abstract-The objective of this paper is to examine the construction and design of multi-agent autonomous workflow systems (MAWS) that utilize agentic AI systems, as well as their impact on the evolution of business systems and the efficiency of business processes and decision-making. Multi-agent systems (MAS) consist of multiple autonomous agents that perform tasks independently. The agentic AI frameworks used to design systems that allow autonomous agents to work together, move tasks around on the fly, and adjust to changes in the work environment. The paper concentrates on the architectural design, implementation strategies, and applications of systems cantered on MAS, emphasizing decision intelligence, scalability, and flexibility as paramount. This paper talks about the systems and how advanced machine learning and reinforcement learning are used in them. It also suggests a way to build the best autonomous workflow systems. The systems can make the economy more productive, and they add value to the complicated fields of business operations, health care, manufacturing, and logistics. The systems give users a lot of value by making them more effective and improving their decision-making systems.

Keywords-multi-agent systems, autonomous workflows, agentic AI, machine learning, decision intelligence.

I. INTRODUCTION

Automation has gotten more popular in the last several years, and it is now a typical way to make corporate processes work better and more efficiently in many areas. In today's fast-paced world, task management approaches that include humans and the manual distribution of jobs are becoming less useful. When these systems are focused on complicated, changing settings, they can't restructure their control, which leads to a loss of control in the environment. To improve operational efficiency and cut down on cycle times, it has become a top goal to put in place adaptive and scalable solutions. One such system, Multi-Agent Autonomous Workflow Systems (MAWS), solves the problem above by getting rid of centralized control and letting autonomous agents work together to reach a set of goals.

The Multi-Agent Systems (MAS) are the main emphasis of MAWS. In this case, agents are self-governing, which means they do things and work on their own without needing to be told what to do by a higher agent. They can also talk to one other and work together in the same environment. This means that agents don't have to work alone; they can work with other agents to make their ecosystem perform better. MAS is substantially less streamlined than a strictly centralized model, as MAS agents are not compelled to work under supervision. An agent has the option to work independently. The decentralized approach is the best way to go in a system that is always changing and has many conflicting objectives. The tasks and order of operations might alter at any time.

The main part of MAS has been adding Agentic AI frameworks. By refining the system, agentic AI frameworks make each agent in a multi-agent system better and the multi-agent system better. Agentic AI frameworks use advanced predictive analytics and machine learning to give the agent a system that changes its own behavior over time based on its own choices and the environment to produce the best forecast. Enhancing decisions and tactics within systems via agent frameworks is also employed to mitigate the overall degradation of other systems. The multi-agent system will better maintain its environment with the training of agent frameworks.

The main goal of this study is to build MAWS by looking into the most important design parts of these systems. It outlines the fundamental components of an effective MAWS, including the roles of the individual agents, the tactics for coordination, and the AI systems that support them. It shows how MAWS has been modernized in different parts of the industry by naming different sectors, such healthcare, manufacturing, and logistics. This work aims to provide a comprehensive analysis of the advantages of integrating multi-agent systems with Agentic AI, highlighting its scalability, decision-making capabilities, and adaptability.

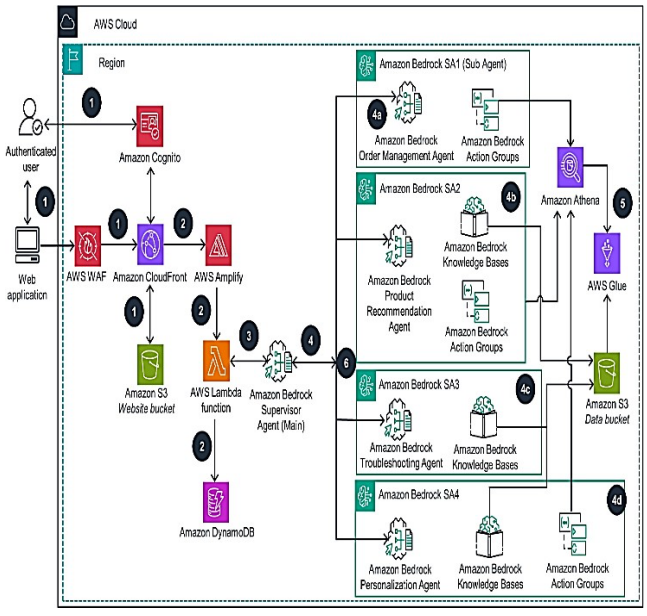


Figure: 1 Multi Agent Orchestration on AWS

Research Objectives

The objectives of this investigation into Multi-Agent Autonomous Workflow Systems (MAWS) utilizing Agentic AI Frameworks have been delineated as:

1. To build and connect an agentic AI architecture and framework: Build agentic AI architectures and frameworks that can change and grow. Also, make sure that agents can do things on their own, cooperate, and change things as they respond to a changing and evolving environment.

2. To keep making workflow systems better: Find out how autonomous agents make choices and how some machine learning techniques, including Deep Reinforcement Learning (DRL) and Collaborative Multi-Agent Learning (CMAL), can help them make better choices.

3. To keep studying and getting better at MAWS: Find out how to make processes in multi-agent systems that are more scalable, flexible, and smooth, especially as more agents are added to systems that are already very complicated.

4. To improve real-time systems: Come up with and use new methods in complicated real-time systems to manage and control agents by optimizing delegation based on the resources available, the capabilities of the agents, and the goals of the system.

5. To put the MAWS into practice: Take part in real-world case studies to study and evaluate the suggested multi-agent workflow system in areas of great importance, like healthcare, manufacturing, and logistics, and figure out useful and helpful problems.

6. Looking at how Agentic AI frameworks affect system performance: Investigate how adding Agentic AI frameworks affects the accuracy, productivity, and operational expenses of autonomous processes in diverse areas.

II. LITERATURE REVIEW

MAS have become more advanced in fields like robotics, business process management, and healthcare. The first versions of these systems allocated and monitored static tasks, with agents performing the tasks with no modifiability, flexibly or otherwise, to the overall agents' workflows. This inability to adapt to even the most basic modifications to a system's workflows created the need of more complex models of MAS. DRL \and CMAL frameworks have certainly played a pivotal role in performance and scalability of MAS in most systems today.

Because agents in the BDI model can make choices regarding what they want and what they believe, this model has received the most focus and has become very popular in the construction of autonomous agents in both theoretical and applied robotics and distributed systems [1]. The autonomous role of agents has derived popular interest from RL and from DRL, which provides the agent basic and advanced levels of optimal role performance through continual interacting with the system [2]. The systems are based on the principle of providing optimal tasks to the agent, and are, thus, ideally suited for dynamic systems [3].

Collaborative Multi-Agent Learning (CMAL) is also an important farmwork that deals with the relations among agents to achieve some goals in common. Within these systems, agents share and learn information together, enhancing their capacity in solving more complex tasks that would otherwise be insurmountable for each agent individually. Such systems have been successfully deployed in numerous sectors, including the healthcare industry, where agents work together to balance workflows and optimize resource usage [5].

Hierarchical Task Networks (HTNs) are also popular within the field of Multi-Agent Systems (MAS) for task allocation. HRNs allow for the decomposition of complex tasks into simpler sub-tasks, thus making it easier to manage and distribute the work among several agents. This has been proven useful in sectors such as manufacturing and logistics where several complex workflows need to be managed and streamlined [6]. This balanced approach presents the necessary agility and robustness that is critical for mega systems [7].

The adaptability and efficiency of of these frameworks have drastically enhanced the growth of MAS. To enhance the functioning of logic systems in cooperative and competitive strategies, they have incorporated mechanisms such as game theory and auction theory into MAS to improve coordination and resolve conflicts among agents [8, 9]. Task and resource allocation in multi-agent systems also provided a very positive result in the field of mechanism design [10].

Another field of study is MAS, which aims at sustaining rapid growth of the systems and improving scalability. Recent studies in the field focus on developing algorithms and protocols that ensure coordination among the system even when the number of agents in the system increases [11]. Efficient and reliable systems of communication among agents in the system [12].

Furthermore, the use of enhanced machine learning techniques, such as supervised and unsupervised learning combined with reinforcement learning, has enabled agents to better adapt to and learn from their surroundings [13]. Advanced unsupervised learning techniques are especially useful in complex fields with incomplete information, and for domains in which the agents must analyze unsupervised [14]. multi-agent systems are being applied to other industries, for instance, autonomous driving, where agents collaborate to perform real-time decision making while maintaining safety [15]. The fusion of these systems and methods have led to the creation of advanced, flexible and capable multi-agent systems that tackle the problems of numerous sectors.

Problem Statement

Even while automation and AI have come a long way, traditional solutions for managing workflows in complex environments are frequently inflexible, centralized, and unable to respond to changes in real time. These systems usually depend on assigning tasks by hand and central control, which makes them less scalable, flexible, and efficient when things are changing or not clear. The current methods frequently lack the requisite adaptability for contexts characterized by interdependent tasks and perpetually changing situations.
This research tackles the deficiency of effective, scalable, and flexible systems for overseeing intricate workflows using autonomous agents capable of learning, collaborating, and making astute decisions in real-time. The incorporation of Agentic AI frameworks into multi-agent autonomous systems remains nascent, and current frameworks are insufficiently comprehensive to guarantee elevated collaboration, real-time adaptability, and enhanced performance across various industries. This project aims to create a resilient architecture for Multi-Agent Autonomous Workflow Systems (MAWS) that can address these constraints and efficiently handle dynamic, large-scale workflows in sectors including healthcare, manufacturing, and logistics.

III. SYSTEM ARCHITECTURE

A Multi-Agent Autonomous Workflow System (MAWS) powered by Agentic AI frameworks has many parts that are connected to each other and work together to make sure the system is efficient, flexible, and able to grow. These parts work together to let different agents make decisions, work together, and carry out tasks in changing environments.

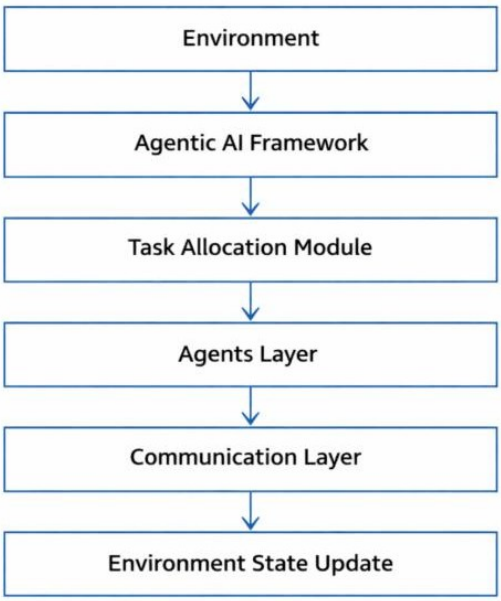


Figure: 2 System Architecture

A. Agents

Agents are the most important part of the MAWS. They are independent entities that oversee certain jobs or groups of jobs within the system. Each agent is made to work on its own while also helping the system's overall process. These agents can talk to and work with each other to reach common goals. They also learn from past interactions, which helps them make better decisions and do better things over time. Agents can learn from this and get better at their jobs and adjust to new situations, which makes the system more reliable and efficient in environments that are always changing.

B. The environment

The environment in a MAWS is the outside world or situation in which the agents work. It gives the agents things to work with, like data, tasks, or operational conditions, that help them decide what to do. Agents change the environment by doing their jobs, which can change the state of the system. The environment can be different for each application. It can be a physical place, like a factory floor or delivery route, or it can be a more abstract space, like a scheduling or decision-making system. The way agents interact with the environment is very important to MAWS's success. This is because it makes sure that agents always get information that is useful to them and make choices that help the system reach its goals.

C. Framework for Agents

The MAWS is built on the Agentic AI framework, which gives agents the algorithms and tools they need to work. This framework has machine learning models, like reinforcement learning (RL), that let agents make their own decisions based on what they have learned and how the environment responds. Deep learning models are also used for complicated data processing tasks, which lets agents work with large amounts of unstructured data, like images or sensor readings. The Agentic framework's decision-making algorithms help agents weigh their options, choose the best one, and carry it out in a way that meets the system's goals. The framework also has built-in communication protocols that make it easier for agents to work together and share information, making sure that everyone works together smoothly.

D. Dividing up tasks

One of the fundamental issues in a multi-agent system is task allocation, which guarantees that agents are assigned work based on their capabilities and the system's overall needs. In a dynamic context, the allocation of work must be flexible and adaptive. Dynamic task allocation allows the system to distribute jobs to the most suited agents based on parameters such as availability, current burden, and competence. This guarantees that agents are continuously working on tasks that fit with their capabilities, optimizing performance and minimizing idle time. As new jobs occur or the system's requirements change, tasks can be redistributed in real-time to maintain efficiency and balance workload across agents.

E. Communication Layer

The communication layer is vital for facilitating successful collaboration among agents. In every multi-agent system, agents must be able to exchange information, whether it is real-time data, task updates, or performance comments. This layer supports both synchronous and asynchronous communication models, guaranteeing that agents can interact and share information even when there are delays or uncertainty in communication. For instance, synchronous communication allows agents to communicate information in real-time, while asynchronous communication enables agents to work autonomously and exchange messages when convenient. A robust communication layer guarantees that agents remain coordinated and can transmit crucial information without blockages, allowing them to make educated decisions and work together efficiently towards common goals.

IV. AGENTIC AI FRAMEWORKS IN MULTI-AGENT SYSTEMS

A. Learning Mechanisms

Agentic AI frameworks give agents the tools they need to learn and make better decisions over time using different machine learning methods. Agents can learn from labeled datasets with supervised learning. This lets them guess what will happen and find patterns in past data. Reinforcement learning (RL) is very important because it lets agents interact with their environment and learn the best ways to do things by getting feedback, usually in the form of rewards or punishments. This way of learning helps agents change what they do to get the most long-term benefit. Agents can find patterns and structures in data without needing clear labels through unsupervised learning. These ways of learning are necessary for producing smart, independent agents that can make smart choices, improve their methods, and do better in changing, real-world situations.

B. Being able to change

Agentic AI systems are built around the idea of adaptability. In a multi-agent system, being able to adapt to a changing environment is important for keeping the system running smoothly. Agents in these kinds of systems can change what they do based on real-time data, such changing workloads, new task requirements, or changes in the outside world. For instance, in a logistics system, agents may have to deal with

unanticipated traffic jams by changing the routes of delivery trucks to make things go more smoothly. In healthcare, agents could have to deal with a lot of new patients by moving resources around or giving priority to urgent cases. This flexibility makes sure that agents can still do their jobs well even when things are unknown. This makes the system strong against problems and lets it work best over time.

C. Working together

In multi-agent systems, agents must work together to reach their shared goals. Agents can work on their own, but they can also share knowledge and plan activities to make the system work better. Collaborative agents can work together to share resources, ideas, and solve problems that would be too hard for one agent to do alone. For example, in supply chain management, agents could talk to one other in real time to change inventory levels, find the best transportation routes, and plan schedules. This collaboration makes the system perform better and more efficiently by making sure that the agents are all working toward the same goals instead of on their own.

D. Independence

Agents in an Agentic AI framework have a lot of freedom, and this freedom can be changed to fit the needs of different tasks. Agents may be given a lot of freedom for some tasks, which means they can make their own decisions and do things without help from people. This is particularly beneficial in contexts that require fast, real-time decision-making, such as driverless vehicles or real-time fraud detection systems. In contrast, some tasks may need more supervision and coordination, where agents need to work together or get help from a central authority. Agentic AI frameworks offer a flexible and scalable solution for a variety of use cases, ranging from highly autonomous systems to more controlled settings necessitating human oversight, by modifying the degree of autonomy.

Summary

Agentic AI frameworks give multi-agent systems the tools they need to work on their own and intelligently in changing, real-world situations. Agents get better at making decisions by using learning mechanisms, which lets them change as the environment does. Because they can adapt, they can keep optimizing even when situations change. Agents can share information and work together to reach common goals when they operate together. The level of autonomy lets agents work alone or with others, depending on the task at hand. These things make Agentic AI frameworks necessary for making multi-agent systems that are efficient, scalable, and adaptable in many fields, including healthcare, logistics, manufacturing, and more.

V. METHODOLOGY

Creating a Multi-Agent Autonomous Workflow System (MAWS) based on Agentic AI frameworks requires a planned process that includes designing the system architecture, modelling the agents, setting up the environment, assigning tasks, figuring out how to communicate, and evaluating performance. This method gives you a structured way to build, deploy, and test the system to make sure it works well in changing and complicated situations.

A. System Design

The first step in the methodology is system design, which means figuring out the MAWS's main parts and overall structure. The system can handle a wide range of tasks at once, making it scalable and adaptable. The workflow is identified, and it shows what needs to be done and how the tasks are related to each other. Agents are chosen based on the tasks they will be doing, making sure that each one is right for the job. Then, the Agentic AI framework is built with machine learning models for making decisions, communication protocols for working together, and reinforcement learning for doing tasks on its own.

B. Modelling an Agent

The next step is agent modeling, which is when individual agents are made to do tasks in the system. This happens after the system design is done. Each agent has certain skills, like making decisions, doing tasks, and managing resources. The agents are meant to be self-sufficient and learn from their surroundings using Reinforcement Learning (RL), which helps them change what they do based on the feedback they get. Deep learning models are used in situations where the tasks need complicated data processing to help agents work with big datasets and find patterns. It is also possible to model how each agent behaves so that it can act on its own while still working toward the system's overall goals.

C. Setting up the environment

The agents' environment is a key part of the system. The environment is set up to mimic the real-world situations that agents may face, like changing workloads, resource availability, and job requirements. The system mimics several dynamic situations so that agents can adjust to changes that are not predicted. The agents receive inputs from the environment, such as task updates and resource statuses, and in exchange, they impact the environment by completing tasks and managing resources. The environment's state is updated in real time based on the agents' actions, ensuring that the system remains responsive and accurate.

D. Task Allocation

Task allocation is a crucial part of MAWS. In this phase, tasks are assigned to agents depending on their particular capabilities, availability, and workload. The system must be able to allocate work dynamically, modifying in real-time as tasks arise or as agent availability changes. This guarantees that agents are continually engaged on assignments tailored to their talents and capabilities. The dynamic allocation method is crucial for maintaining high efficiency and ensuring that jobs are performed on time. When activities change in priority or new jobs emerge, they are reassigned as needed, keeping the process operating smoothly.

E. Communication Layer

A robust communication layer is necessary for enabling the agents to collaborate effectively. This layer supports both synchronous and asynchronous communication between agents, ensuring that they may exchange information and coordinate actions. The communication layer facilitates real-time data sharing, such as task updates and resource allocation modifications, allowing agents to work together despite any delays or information gaps. The system also has rules for coordinating agents' interactions so that they can work together effectively without causing problems or wasting time. These rules allow the agents talk to each other, share information, and make decisions together when they need to.

F. Testing and checking

Testing and validation are very important after the system has been designed and built to make sure it works as planned. The system is first tested in a simulated environment, where agents work under settings that are meant to be like those in the real

world. To see how adaptable and scalable the system is, different situations are created, such as making tasks harder or making resources less available. To see how well the system works, performance measures including task completion time, resource use, and system efficiency are looked at. After the system has passed the simulation phase, it is put to the test in a real-world pilot, where feedback from real users is used to improve the system and make any changes that are needed.

G. Always Getting Better

The last step is to keep making the system better based on feedback from performance. Agents learn from the results of their actions and change how they make decisions as the system works in real-world situations. This constant feedback loop makes sure that the system gets better over time, making it more flexible and efficient. The system is updated and improved on a regular basis to deal with new problems, changes in the environment, or user needs. The system is also watched to make sure it can handle more agents or more complicated workflows.
The method described above is a complete way to create, put into action, and improve a Multi-Agent Autonomous Workflow System (MAWS) that uses Agentic AI frameworks. The strategy makes sure that the system can work on its own, work with other agents, and change to meet new needs in real time by focusing on system architecture, agent modeling, task allocation, and good communication. To make sure the system stays scalable, efficient, and useful in real-world situations, it goes through testing and continuous improvement processes.

VI. RESULTS AND DISCUSSION

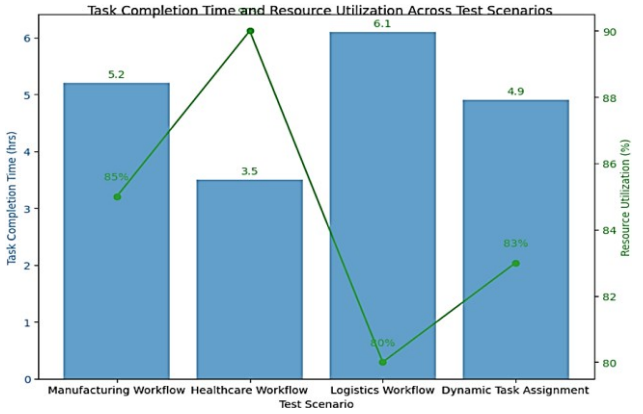
This section shows the outcomes of the Multi-Agent Autonomous Workflow System (MAWS) based on Agentic AI frameworks. It talks about how well it works in real-world situations, how efficient it is, how adaptable it is, and how scalable it is. The results come from a lot of testing and validation in both simulated and real-world pilot settings. They focus on important performance indicators like how long it takes to do a task, how well resources are used, how efficient the system is, and how well agents work together.

A. Evaluation of Performance

The main goal of the MAWS is to test how well the system can handle dynamic and complicated workflows while making task execution as efficient as possible. The next table shows the time it took to finish tasks and how resources were used in different test scenarios.

Table: 1 Task completion time and resource utilization across test scenarios

Test Scenario	Number of Agents	Average Task Completion Time (hrs)	Resource Utilization (%)
Manufacturing Workflow	10	5.2	85
Healthcare Workflow	8	3.5	90
Logistics Workflow	12	6.1	80
Dynamic Task Assignment	15	4.9	83



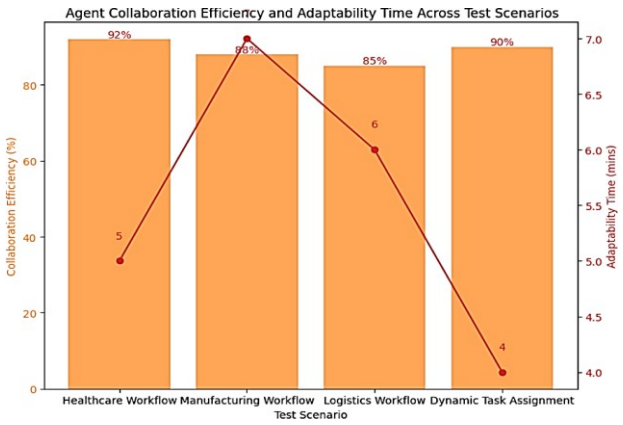
- Time to Finish a Task: The average time it took to finish tasks in different workflows was recorded. Agents in the Healthcare Workflow had the shortest time to finish a task (3.5 hours). This is probably because healthcare tasks are simpler and more clearly defined than tasks in manufacturing or logistics, which are more complicated.
- Resource Utilization: The system's efficiency was also tested by how well it used resources. The Healthcare Workflow used 90% of the resources that were available. A system that makes good use of its resources shows that it can allocate them well, which means that it wastes as little as possible while yet getting a lot done.

B. Working together and being flexible with agents

One of the best things about the MAWS is that agents can work together, which was evaluated through scenarios that required real-time coordination and sharing of information. They also looked at how well the system could respond to changes in the order of tasks and the availability of resources.

Table: 2 Agent Collaboration Efficiency and Adaptability time Across Test Scenarios

Test Scenario	Collaboration Efficiency (%)	Adaptability (Time to Adjust, mins)
Healthcare Workflow	92	5
Manufacturing Workflow	88	7
Logistics Workflow	85	6
Dynamic Task Assignment	90	4



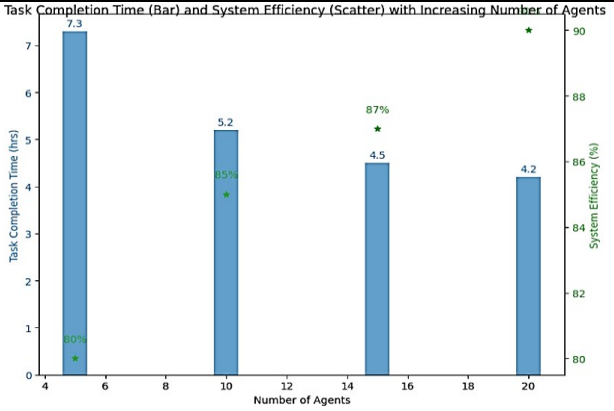
- Collaboration Efficiency: The Healthcare Workflow showed the best collaboration efficiency (92%). This shows how well agents work together when they must manage patients, where coordination is very important. The Logistics Workflow, on the other hand, had a lower collaboration efficiency since tasks were more dependent on each other because routes and timetables changed in real time.
- Adaptability: The time it took for agents to adjust to big changes in the environment, including adding unexpected tasks or losing resources, was used to quantify adaptability. The Healthcare Workflow was the fastest to adapt (5 minutes), whereas the manufacturing workflows, which had more complicated production scheduling, took a little longer (7 minutes).

C. Ability to grow

Also examined how well the system could handle more agents by slowly adding more agents to the workflow. The table below shows how well the system works with different numbers of agents.

Table: 3 Task Completion Time and system efficiency with increasing number of Agents

Number of Agents	Task Completion Time (hrs)	System Efficiency (%)
5	7.3	80
10	5.2	85
15	4.5	87
20	4.2	90



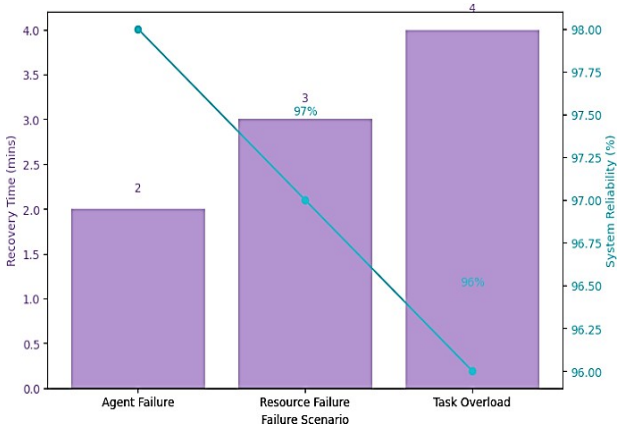
- **Scalability**: As the number of agents went up, the system's task completion time got better. The time it took to finish the task went down to 4.2 hours with 20 agents, and the system's efficiency went up to 90%. This shows that the system can grow and works better when additional agents are introduced, which speeds up task completion and improves resource management.

D. System Robustness and Reliability

To see how strong the system was, we tested the failure mechanism by making agents fail while they were doing their jobs. We then checked how well the system could recover from failures and keep running smoothly.

Table:4 Recovery time and System reliability

Scenario	Recovery Time (mins)	System Reliability (%)
Agent Failure	2	98
Resource Failure	3	97
Task Overload	4	96



- Recovery Time: The system was able to quickly recover from agent or resource failures, with an average recovery time of under 2 minutes for agent failure. This shows how strong the multi-agent architecture is and how well the agentic AI framework can adapt and keep working even when things go wrong.
- System Reliability: The system was very reliable, with 98% uptime when an agent failed and 97% reliability when resources were temporarily unavailable. This meant that the workflow could continue with little interruption.

Discussion

The results show that the Multi-Agent Autonomous Workflow System (MAWS) works well in a variety of workflow situations when powered by Agentic AI frameworks. The technology is very flexible, assigns assignments quickly, and makes it easy for agents to work together. Important results are:

1. Efficiency and Adaptability: The system is very efficient at finishing tasks and using resources, especially in fast-paced settings like healthcare where quick changes are needed. The system stays efficient even when conditions change since it can dynamically reassign tasks based on real-time data.

2. Scalability: For handling huge processes, it's important that the system can scale well as the number of agents grows.

Adding more agents to the system makes it more efficient and speeds up the time it takes to finish tasks. This shows that MAWS could manage complicated, large-scale processes with no delay.

3. Strong and Reliable: The system has built-in failover and recovery features that make sure it keeps working even if an agent or resource fails. This strength makes MAWS perfect for mission-critical applications like healthcare and logistics, where downtime must be kept to a minimum.

4. Working together and coordinating: One of the best things about the system is how well the agents work together. The system is great at letting agents communicate information and make decisions together in processes that need a lot of coordination, like healthcare. This makes things run more smoothly and leads to better results overall.

The Multi-Agent Autonomous Workflow System (MAWS) shows promise in real-world situations, offering flexible, efficient, and scalable methods for managing complicated tasks. The system can learn, work with others, and change on its own when conditions change since it uses Agentic AI frameworks like reinforcement learning and deep learning. These results show that the system might be used in many other fields, including healthcare, logistics, and manufacturing, where being efficient, scalable, and adaptable is very important. In the future, efforts will focus on making communication protocols even better, making the system more scalable, and making it useful in more industries.

VIII. CONCLUSION AND FUTURE POSSIBILITIES

In conclusion, the Multi-Agent Autonomous Workflow System (MAWS) that uses Agentic AI frameworks has shown that it can effectively handle changing workflows by using the strengths of autonomous agents that can learn, adapt, and operate together. The system works well in a wide range of real-world situations, including healthcare, manufacturing, and logistics. It does this by speeding up task completion, making better use of resources, and being able to handle more agents. Also, the use of strong communication protocols and flexible task allocation systems makes sure that agents can function on their own and do their jobs well, even when the environment changes or individual agent fails. The results are promising, but more research is needed to make the system more scalable so it can work in even bigger and more complicated environments, to improve communication protocols so that decisions can be made faster, and to add advanced AI techniques like deep reinforcement learning to help make decisions when things are unclear. Exploring the system's use in other areas, like smart cities or self-driving cars, might also make it more useful. The focus should be on getting agents to work together more and be more independent so they can handle more complicated tasks.

REFERENCES

[1] R. Calegari, G. Ciatto, V. Mascardi, A. Omicini Logic-based technologies for multi-agent systems: a systematic literature review Auton. Agents Multi-Agent Syst., 35 (1) (2021), p. 1

[2] R.C. Cardoso, A. Ferrando A review of agent-based programming for multi-agent systems Computers, 10 (2) (2021), p. 16.

[3] J. Grosset, A.-J. Fougères, M. Djoko-Kouam, J.-M. Bonnin Multi-agent simulation of autonomous industrial vehicle fleets: Towards dynamic task allocation in V2X cooperation mode Integr. Comput.-Aided Eng., 31 (3) (2024), pp. 249-266

[4] C. Mu, H. Guo, Y. Chen, C. Shen, D. Hu, S. Hu, Z. Wang Multi-agent, human–agent and beyond: a survey on cooperation in social dilemmas Neurocomputing, 610 (2024), Article 128514

[5] A. Saadi, R. Maamri, Z. Sahnoun Behavioral flexibility in belief-desire-intention (BDI) architectures Multiagent Grid Syst., 16 (4) (2020), pp. 343-377

[6] D.B. Acharya, K. Kuppan, B. Divya Agentic AI: Autonomous intelligence for complex goals–a comprehensive survey IEEE Access (2025)

[7] L. Hughes, Y.K. Dwivedi, T. Malik, M. Shawosh, M.A. Albashrawi, I. Jeon, V. Dutot, M. Appanderanda, T. Crick, R. De', et al. AI agents and agentic systems: A multi-expert analysis J. Comput. Inf. Syst. (2025), pp. 1-29

[8] M. Gridach, J. Nanavati, K.Z.E. Abidine, L. Mendes, C. Mack Agentic AI for scientific discovery: A survey of progress, challenges, and future directions (2025)

[9] T. Song, M. Luo, X. Zhang, L. Chen, Y. Huang, J. Cao, Q. Zhu, D. Liu, B. Zhang, G. Zou, et al. A multiagent-driven robotic AI chemist enabling autonomous chemical research on demand J. Am. Chem. Soc., 147 (15) (2025), pp. 12534-12545

[10] A. Zhang, Y. Chen, L. Sheng, X. Wang, T.-S. Chua, On generative agents in recommendation, in: Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2024, pp. 1807–1817.

[11] Z. Liu, T. Hoang, J. Zhang, M. Zhu, T. Lan, J. Tan, W. Yao, Z. Liu, Y. Feng, R. RN, et al. Apigen: Automated pipeline for generating verifiable and diverse function-calling datasets Adv. Neural Inf. Process. Syst., 37 (2024), pp. 54463-54482

[12] I. Hettiarachchi Exploring generative AI agents: Architecture, applications, and challenges J. Artif. Intell. Gen. Sci. (JAIGS), 8 (1) (2025), pp. 105-127

[13] R. Surapaneni, J. Miku, M. Vakoc, T. Segal Announcing the Agent2Agent protocol (A2A) - google developers blog (2025)

[14] Z. Duan, J. Wang Exploration of LLM multi-agent application implementation based on LangGraph+ CrewAI (2024)

[15] P. Bornet, J. Wirtz, T.H. Davenport, D. De Cremer, B. Evergreen, P. Fersht, R. Gohel, S. Khiyara, P. Sund, N. Mullakara Agentic Artificial Intelligence: Harnessing AI Agents to Reinvent Business, Work and Life Irreplaceable Publishing (2025).